

 高等教育规划教材

Visual Basic 程序设计教程

第 3 版

刘瑞新 汪远征 主编



提供电子教案和习题解答

下载网址 <http://www.cmpedu.com>



机械工业出版社
CHINA MACHINE PRESS

高等教育规划教材

Visual Basic 程序设计教程

第3版

刘瑞新 汪远征 主 编
徐雅静 汪晓诗 由赢公 等编著



机械工业出版社

本书以 Visual Basic 6.0 中文版为语言背景,以程序结构为主线,采用案例方式,通过大量实例,全面、细致地讲解了 Visual Basic 可视化面向对象编程的概念和方法。主要内容包括 Visual Basic 程序设计概述、Visual Basic 语言基础、数据的输入与输出、选择结构程序设计、循环结构程序设计、数组、过程、变量与过程的作用范围、菜单与对话框、多重窗体与环境应用、键盘与鼠标事件过程、图形与图像、数据文件、面向对象的程序设计和数据库访问技术。本书涵盖了最新《全国计算机等级考试二级考试大纲(Visual Basic 程序设计)》的内容。全书概念清楚、逻辑性强、层次分明、例题丰富,适合教师课堂教学和学生自学。

本书适合作为大学本科、高职高专院校的教材,也适合作为全国计算机等级考试 Visual Basic 程序设计二级考试的教材。

本书配有电子教案,需要的教师可登录 www.cmpedu.com 免费注册,审核通过后下载,或联系编辑索取(QQ: 2966938356, 电话: 010-88379739)。

图书在版编目(CIP)数据

Visual Basic 程序设计教程 / 刘瑞新, 汪远征主编. —3 版. —北京: 机械工业出版社, 2015.1

高等教育规划教材

ISBN 978-7-111-48279-6

I. ①V… II. ①刘… ②汪… III. ①BASIC 语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2014)第 238418 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

责任编辑: 和庆娣 责任校对: 张艳霞

责任印制: 李 洋

北京瑞德印刷有限公司印刷(三河市胜利装订厂装订)

2015 年 1 月第 3 版·第 1 次印刷

184mm×260mm·20 印张·496 千字

0001—3000 册

标准书号: ISBN 978-7-111-48279-6

定价: 39.90 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

电话服务

网络服务

服务咨询热线: (010) 88379833

机工官网: www.cmpbook.com

读者购书热线: (010) 88379649

机工官博: weibo.com/cmp1952

教育服务网: www.cmpedu.com

封面无防伪标均为盗版

金书网: www.golden-book.com

出版说明

当前,我国正处在加快转变经济发展方式、推动产业转型升级的关键时期。为经济转型升级提供高层次人才,是高等院校最重要的历史使命和战略任务之一。高等教育要培养基础性、学术型人才,但更重要的是加大力度培养多规格、多样化的应用型、复合型人才。

为顺应高等教育迅猛发展的趋势,配合高等院校的教学改革,满足高质量高校教材的迫切需求,机械工业出版社邀请了全国多所高等院校的专家、一线教师及教务部门,通过充分的调研和讨论,针对相关课程的特点,总结教学中的实践经验,组织出版了这套“高等教育规划教材”。

本套教材具有以下特点:

- 1) 符合高等院校各专业人才的培养目标及课程体系的设置,注重培养学生的应用能力,加大案例篇幅或实训内容,强调知识、能力与素质的综合训练。
- 2) 针对多数学生的学习特点,采用通俗易懂的方法讲解知识,逻辑性强、层次分明、叙述准确而精炼、图文并茂,使学生可以快速掌握,学以致用。
- 3) 凝结一线骨干教师的课程改革和教学研究成果,融合先进的教学理念,在教学内容和方法上做出创新。
- 4) 为了体现建设“立体化”精品教材的宗旨,本套教材为主干课程配备了电子教案、学习与上机指导、习题解答、源代码或源程序、教学大纲、课程设计和毕业设计指导等资源。
- 5) 注重教材的实用性、通用性,适合各类高等院校、高等职业学校及相关院校的教学,也可作为各类培训班教材和自学用书。

欢迎教育界的专家和老师提出宝贵的意见和建议。衷心感谢广大教育工作者和读者的支持与帮助!

机械工业出版社

前 言

本书第1版特色显著,深受广大师生的欢迎,连续多年被许多高等院校选为教材。但随着 Visual Basic 新版本的推出(Visual Basic .NET、Visual Basic 2005~2013), Visual Basic 6.0 现在已经很少作为实际的程序设计开发工具,而演变成为一种相关专业程序设计方法的教学语言和全国计算机等级考试的考试课程,所以目前许多大专院校的程序设计课程讲授的仍然是 Visual Basic 6.0。为了适应新的教学要求,我们编写了第2版。在编写第2版时,为了使學生掌握更多的程序设计知识,我们把重点放在程序设计的层面上,而不是某个程序设计语言上。为此,在教材的组织上,以可视化程序设计作为主线,以相关的 Visual Basic 控件作为辅助,使學生重点掌握程序设计的基本知识、设计思想以及可视化程序设计的通用方法与步骤,这样學生在学习本课程时就不再局限在 Visual Basic 6.0 上,在以后需要编程解决实际问题时,可以很容易地过渡到其他程序设计语言(如 Visual Basic 2012、Visual C#、Java 等)。

在内容具体细节的安排上,本书把难点分散到各章节中,采用案例方式,每章均以具有代表性、实用性、趣味性的实例贯穿其中,使學生容易理解,提高分析问题和解决问题的能力,掌握 Windows 环境中的可视化面向对象编程的概念和方法,包括 Visual Basic 程序设计概述、Visual Basic 语言基础、数据的输入与输出、选择结构程序设计、循环结构程序设计、数组、过程、变量与过程的作用范围、菜单与对话框、多重窗体与环境应用、键盘与鼠标事件过程、图形与图像、数据文件、面向对象的程序设计和数据库访问技术等。本书的教学时数建议为 72 学时(其中授课为 36 学时,上机练习为 36 学时)。

作为本书的第3版,我们修正了一些疏漏,按照最新的《全国计算机等级考试二级考试大纲(Visual Basic 程序设计)》增加了近几年全国计算机等级考试二级 Visual Basic 的考试真题作为例题或习题,还在第2版的基础上增加了“图形与图像”一章,而删去“Visual Studio 2005 简介”一章,以便让学生更加专注于 Visual Basic 内容的学习,学习本书后,可参加全国计算机等级考试。

本书由刘瑞新、汪远征主编,徐雅静、汪晓诗等编著,刘瑞新编写第1、15章,汪远征编写第2、5、8、9章,由赢公编写第3章,徐雅静编写第4、7、10章,贺俊华编写第6章,汪晓诗编写第11、12、13章,第14章及试题的验证、电子教案制作等由刘克纯、陈文焕、刘有荣、李刚、田金雨、曹媚珠、骆秋容、王如雪、孙明建、李索、刘大学、缪丽丽、沙世雁、田金凤、陈文娟、王茹霞、田同福、徐维维、徐云林、李继臣、王如新、赵艳波编写完成,本书由刘瑞新、汪远征统编定稿。

计算机技术发展迅速,书中难免存在疏漏和不足,欢迎读者批评指正。同时也希望老师和同学们提出宝贵意见,以便完善本书的教材体系。

编 者

目 录

出版说明

前言

第1章 Visual Basic 程序设计概述

1.1 计算机程序设计概述

1.1.1 程序设计语言简介

1.1.2 算法及其描述

1.1.3 程序设计方法概述

1.2 Visual Basic 简介

1.2.1 Visual Basic 概述

1.2.2 Visual Basic 的发展过程

1.2.3 Visual Basic 的特点

1.3 Visual Basic 程序设计的基本

概念

1.3.1 可视化编程与事件驱动编程

1.3.2 对象的属性、事件和方法

1.3.3 Visual Basic 的编程环境

1.3.4 Visual Basic 的窗体与控件

1.4 Visual Basic 编程初步

1.4.1 Visual Basic 可视化编程的步骤

1.4.2 控件的画法

1.4.3 公共的属性与事件过程

1.5 习题

第2章 Visual Basic 语言基础

2.1 标准数据类型

2.1.1 数值(Numeric)型数据

2.1.2 字符(String)型数据

2.1.3 布尔(Boolean)型数据

2.1.4 日期(Date)型数据

2.1.5 对象(Object)型数据

2.1.6 可变(Variant)类型数据

2.2 变量

2.2.1 变量的命名规则

2.2.2 变量的声明

2.3 常量

2.3.1 直接常量

2.3.2 符号常量

2.4 表达式

2.4.1 算术表达式

2.4.2 字符串表达式

2.4.3 日期表达式

2.5 常用内部函数

2.5.1 数学运算函数

2.5.2 字符串函数

2.5.3 日期和时间函数

2.5.4 格式输出函数

2.5.5 随机数语句和函数

2.5.6 数据类型转换函数

2.6 程序语句

2.6.1 语句

2.6.2 语句的书写规则

2.6.3 命令格式中的符号约定

2.6.4 基本语句

2.7 习题

第3章 数据的输入与输出

3.1 数据输出

3.1.1 Print 方法

3.1.2 信息框函数 MsgBox

3.1.3 使用标签控件

3.2 数据输入

3.2.1 使用“文本框”控件

3.2.2 输入框函数 InputBox

3.3 打印机输出

3.3.1 直接输出

3.3.2 窗体输出

3.4 使用框架控件

3.5 习题

第4章 选择结构程序设计	78	6.1.5 数组的声明	131
4.1 选择结构程序设计的概念	78	6.2 数组的基本操作	132
4.2 条件表达式	78	6.2.1 数组元素的输入、输出和复制	132
4.2.1 关系运算符与关系表达式	78	6.2.2 For Each...Next 语句	134
4.2.2 布尔运算符与布尔表达式	79	6.2.3 数组的初始化	135
4.2.3 运算符的优先顺序	80	6.2.4 数组的使用	136
4.3 单条件选择语句 If	81	6.3 动态数组	141
4.3.1 单行结构条件语句 If...Then...Else	81	6.3.1 创建动态数组	141
4.3.2 使用 If 函数	83	6.3.2 保留动态数组的内容	143
4.3.3 块结构条件语句 If...Then...Else... End If	84	6.4 控件数组	144
4.3.4 If 语句的嵌套	85	6.4.1 控件数组的概念	144
4.4 多分支条件选择语句		6.4.2 控件数组的建立	144
Select Case	88	6.4.3 控件数组的使用	145
4.4.1 Select Case 结构与语法	88	6.5 习题	148
4.4.2 Select Case 结构的应用	89	第7章 过程	153
4.5 使用计时器控件	92	7.1 Sub 过程	153
4.6 提供简单选择的控件	93	7.1.1 事件过程与通用过程	153
4.6.1 选项按钮	93	7.1.2 通用过程的创建	154
4.6.2 复选框	96	7.1.3 通用过程的调用	156
4.7 习题	100	7.2 Function 过程	158
第5章 循环结构程序设计	105	7.2.1 Function 过程的定义	158
5.1 循环结构程序设计的概念	105	7.2.2 Function 过程的调用	159
5.2 Do...Loop 语句	105	7.2.3 查看过程	161
5.2.1 前测型 Do...Loop 循环	105	7.3 向过程传递参数	161
5.2.2 后测型 Do...Loop 循环	108	7.3.1 形式参数与实际参数	162
5.3 For...Next 语句	110	7.3.2 按值传递与按地址传递	162
5.4 列表框与组合框	112	7.3.3 使用参数	164
5.4.1 列表框	113	7.3.4 传递数组	166
5.4.2 组合框	118	7.4 过程的嵌套与递归调用	169
5.5 算法举例	121	7.4.1 过程的嵌套	169
5.6 习题	123	7.4.2 过程的递归	170
第6章 数组	130	7.5 习题	171
6.1 数组的概念	130	第8章 变量与过程的作用范围	178
6.1.1 数组与数组元素	130	8.1 代码模块的概念	178
6.1.2 数组的类型	130	8.1.1 窗体模块	178
6.1.3 数组的维数	130	8.1.2 标准模块	179
6.1.4 数组的形式	131	8.1.3 类模块	180
		8.2 变量的作用范围	180

8.2.1 过程级变量	180	第 12 章 图形与图像	220
8.2.2 模块级变量	181	12.1 绘制图形	220
8.2.3 变量的生存期	182	12.1.1 图形控件	220
8.3 过程的作用范围	183	12.1.2 图形的坐标系统	223
8.3.1 模块级过程	184	12.1.3 与图形有关的属性	224
8.3.2 全局级过程	184	12.1.4 使用颜色	227
8.3.3 调用其他模块中的过程	184	12.1.5 常用绘图方法	228
8.4 习题	186	12.1.6 绘图语句与 Paint 事件	233
第 9 章 菜单与对话框	187	12.2 显示图片	233
9.1 使用菜单	187	12.2.1 直接加载图片到窗体	234
9.1.1 下拉式菜单	187	12.2.2 使用图像控件	234
9.1.2 弹出式菜单	193	12.2.3 使用图片框控件	236
9.2 公共对话框	194	12.3 滚动条控件	240
9.2.1 添加“公共对话框”控件	195	12.3.1 滚动条的属性	240
9.2.2 使用“公共对话框”	195	12.3.2 滚动条的事件	241
9.3 习题	200	12.3.3 滚动条的应用	241
第 10 章 多重窗体与环境应用	203	12.4 习题	244
10.1 多重窗体与多文档窗体	203	第 13 章 数据文件	246
10.1.1 建立多重窗体应用程序	203	13.1 文件的结构与分类	246
10.1.2 多重窗体程序的执行	204	13.1.1 文件的分类	246
10.1.3 Sub Main 过程	205	13.1.2 文件的结构	247
10.1.4 多重窗体程序应用	206	13.2 文件操作语句和函数	247
10.1.5 多文档 (MDI) 窗体	207	13.2.1 数据文件的操作	247
10.2 DoEvents 函数与闲置循环	208	13.2.2 文件的打开与关闭语句	247
10.2.1 DoEvents 函数	208	13.2.3 文件访问函数	249
10.2.2 闲置循环	209	13.3 顺序文件的操作	251
10.3 习题	210	13.3.1 顺序文件的写操作	251
第 11 章 键盘与鼠标事件过程	211	13.3.2 顺序文件的读操作	253
11.1 键盘事件	211	13.4 随机文件的操作	259
11.1.1 KeyPress 事件	211	13.4.1 随机文件的读/写操作	260
11.1.2 KeyDown 和 KeyUp 事件	212	13.4.2 随机文件中记录的增加与删除	262
11.1.3 使用 KeyPreview 属性	214	13.5 文件系统控件	264
11.2 鼠标事件	214	13.5.1 驱动器列表框	264
11.2.1 MouseDown 和 MouseUp 事件	214	13.5.2 目录列表框	264
11.2.2 MouseMove 事件	215	13.5.3 文件列表框	265
11.2.3 自定义鼠标指针	215	13.5.4 文件系统控件共有的属性	266
11.2.4 使用鼠标事件	216	13.5.5 文件系统对象的同步操作	266
11.3 习题	219	13.6 文件基本操作	267

13.6.1	目录的基本操作	267
13.6.2	文件的基本操作	268
13.7	习题	269
第 14 章	面向对象的程序设计	272
14.1	面向对象程序设计概述	272
14.1.1	对象和类	272
14.1.2	类的继承性	273
14.1.3	类的封装性	273
14.1.4	类的多态性	274
14.2	Visual Basic 中预定义的和对象	274
14.2.1	通过控件类创建对象	274
14.2.2	通过代码创建对象	274
14.3	在应用程序中创建和使用类	276
14.3.1	创建和使用自定义类	276
14.3.2	向类中添加属性	278
14.3.3	向类中添加方法	279
14.3.4	响应默认事件和自定义事件	281

14.4	ActiveX 控件	282
14.4.1	ActiveX 控件概述	283
14.4.2	添加删除 ActiveX 控件	283
14.4.3	创建 ActiveX 控件	283
14.5	习题	288
第 15 章	数据库访问技术	290
15.1	Access 2003 数据库	290
15.2	使用数据控件	291
15.2.1	数据控件的属性	291
15.2.2	数据控件的事件	292
15.2.3	数据控件的方法	293
15.2.4	记录集对象 (Recordset)	294
15.3	使用 ADO 控件	301
15.3.1	ADO 数据控件的属性、方法和事件	302
15.3.2	高级数据绑定控件	305
15.3.3	使用数据窗体向导	310
15.4	习题	311

第1章 Visual Basic 程序设计概述

本章主要介绍程序设计的一些基本知识、基本概念和基本方法，为读者学习和掌握 Visual Basic（简称 VB）程序设计奠定基础。

1.1 计算机程序设计概述

计算机程序就是计算机解决某些特定问题所需的符号化指令序列，或者说是用计算机语言描述的特定问题的解决步骤。编写程序的过程称为程序设计，在程序设计时离不开程序设计语言。通常把给计算机编制程序的符号系统及规则称为计算机的程序设计语言。这些符号系统及规则构成了计算机的指令系统。当人们利用计算机完成一项工作时，只需要告诉计算机什么时候、在什么条件下干什么，计算机便根据指令一条一条地执行，并返回结果。

1.1.1 程序设计语言简介

任何一个计算机系统都是按照人们用某种程序设计语言编写的程序进行工作的，人们通过程序设计语言编写的程序来指挥和控制计算机运行。程序设计语言是人与计算机进行交流的有效工具，在计算机科学技术的发展过程中，发挥了巨大的作用。

程序设计语言的产生和发展，直接推动了计算机的普及和应用。在计算机不断发展的历史过程中，程序设计语言也经历了从低级到高级的发展阶段。

1. 机器语言

计算机能直接识别的程序设计语言只有机器语言。机器语言是计算机能执行的指令代码，这种语言是由若干 0 和 1 的序列组成的指令，也就是人们常说的二进制代码。用机器语言设计的程序称为机器语言程序，这是一种最低级的计算机语言程序。由于这种程序全部由二进制数字组成，所以难记、难写、难读，而且在程序设计过程中很容易出错，一旦出错也不容易检查。机器语言难记、难写、难读的特点，使机器语言程序维护起来困难重重。另外，由于不同类型的机器，其二进制代码系统也不相同，所以在一台计算机上设计的程序，到另一台计算机上往往无法使用，从而使程序的可移植性很差。总之，由于机器语言的上述特点，使得用这种语言设计程序效率低，操作困难，不利于程序设计的推广与应用。

2. 汇编语言

针对机器语言的上述特点，人们对机器语言进行了改进，使用一种比较直观、便于记忆的指令符号来代替二进制数字的机器指令代码，这就是汇编语言。汇编语言的每条指令通常使用英文单词或其缩写形式表示，也叫助记符，例如，用 ADD（英文单词“加”）表示加，用 SUB（英文单词“减”subtract 的缩写）表示减等。助记符相对于二进制数字的机器指令代码来说容易记忆，所以汇编语言的出现，是程序设计语言的一大进步，甚至可以说，汇编语言是高级语言的前驱。

汇编语言使用的助记符不是二进制的机器代码语言，因此计算机无法识别，但是汇编语言的这些助记符与机器指令代码是一一对应的，只需用一个专门的程序将其转换为机器指令代码

即可，因此用汇编语言设计的程序，与机器语言程序运行的速度相仿，而这个负责转换的程序叫汇编程序。转换前的汇编程序叫源程序，转换后的汇编程序叫目标程序。

不过，汇编语言相对于人们熟悉的自然语言仍然相差较远，而且汇编语言与机器语言相同，它也是面向机器的。换句话说，用汇编语言在一台机器上设计的程序，到另一机器上往往不能运行，即可移植性差。

由于机器语言与汇编语言都是面向机器的，所以人们也叫它们低级语言。

3. 高级语言

20 世纪 50 年代中期出现了高级语言，高级语言比较圆满地克服了机器语言与汇编语言的不足，是程序设计语言的一大突破。高级语言接近人们熟悉的自然语言（主要指英语），掌握与使用都十分方便。高级语言具有通用性，在其初始阶段是面向过程的语言。高级语言与具体的计算机指令系统没有直接关系，因此用高级语言设计的程序可以在各种类型的计算机上运行。当然，计算机并不能识别与执行用高级语言设计的程序，因此，必须将高级语言程序转换为机器语言程序，才能在计算机上得以执行，这种转换的过程叫“翻译”。

任何一种高级语言系统都包含专门用于“翻译”的程序。对高级语言的“翻译”有两种方式，一种是“解释”方式，即“翻译”一句执行一句，负责这种“翻译”方式的程序叫解释程序；另一种是“编译”方式，是将整个程序“翻译”完毕后再予以执行，负责这种“翻译”方式的程序叫编译程序。

“翻译”前的程序叫源程序或源代码，源代码通常是文本形式；“翻译”后的程序叫目标程序或目标代码，目标代码是二进制形式。

用解释程序“翻译”执行程序比编译程序“翻译”执行程序慢得多。不论是解释还是编译，在“翻译”过程中都会自动检查源程序中的语法错误。

1.1.2 算法及其描述

1. 算法的概念

什么是算法？当代著名计算机科学家 D. E. Knuth 称：“一个算法，就是一个有穷规则（指令）的集合。其中之规则规定了一个解决某一特定类型的问题的运算序列。”简单地说，任何解决问题的过程都是由一定的步骤组成的，把解决问题确定的方法和有限的步骤称作算法。

程序设计主要包括两个方面：行为设计与结构设计。行为设计是对要解决的问题提出达到目的需要实施的一些步骤，并对这些步骤加以必要的细化，在此基础上用某些方式完整地描述出来，其结果就是算法；结构设计是针对所要解决的问题，对数据定义数据结构（包括物理结构和逻辑结构）。有了好的算法、合适的数据结构，再使用某些程序设计语言加以具体实现，即可得到程序。

凡是有过程序设计经历的人都会对 N. Wirth 提出的“算法+数据结构=程序”有深刻的领悟。算法是程序的灵魂，它在产生程序的过程中占有重要的地位。

通俗地说，算法就是指为解决一个问题而采取的方法和步骤，或者说是解题步骤的精确描述。不要认为只有“计算”问题才有算法。广义地说，处理任何问题都有一个“算法”问题，例如，菜谱就包含算法，因为它除了列出做菜的原料以外，还列出操作的每一步骤。当然，这里讨论的是计算机算法，即计算机能执行的算法。

2. 算法的表示

表示一个算法可以采用不同形式。

(1) 用自然语言表示算法

可以用人们日常生活中使用的语言即自然语言来表示算法，用自然语言表示算法的好处是人人都懂，人人都会。

【例 1-1】计算函数 $M(x)$ 的值：

$$M(x) = \begin{cases} bx + 2a & x \leq a \\ a(c - x) + 3c & x > a \end{cases}$$

其中 a, b, c 为常数。

算法分析：本题是一个数值运算问题。其中 M 代表要计算的函数值，有两个不同的算式，根据 x 的取值决定采用哪一个算式。根据计算机具有逻辑判断的基本功能，用计算机解题的算法如下：

- 1) 将 a, b, c 和 x 的值输入到计算机。
- 2) 判断 $x \leq a$ ，如果条件成立，执行步骤 3)，否则执行步骤 4)。
- 3) 按表达式 $bx + 2a$ 计算出结果存放到 M 中，然后执行步骤 5)。
- 4) 按表达式 $a(c - x) + 3c$ 计算出结果存放到 M 中，然后执行步骤 5)。
- 5) 输出 M 的值。
- 6) 算法结束。

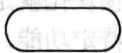
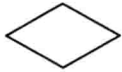
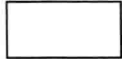
上面的过程就是算法，即解决问题的方法与步骤，可以看出，用自然语言表示清楚、易懂，但往往需要非常冗长的文字来表述。而且用自然语言来表示算法有时会产生“二义性”，例如，“张三对李四说他的儿子考上了大学”，就使人弄不明白是张三的儿子还是李四的儿子。当然在现实生活中，这种二义性较容易解决，比如说张三只有女儿。但在计算机编程中，这种二义性是无法解决的。因此除了简单的问题外，通常不用自然语言表示算法。

(2) 用流程图表示算法

流程图是用一些图框、流程线以及文字说明来描述解决问题的方法与步骤。用流程图来表示算法，直观、形象、容易理解。

美国国家标准化协会（American National Standard Institute, ANSI）规定了一些常用的流程图符号，见表 1-1。

表 1-1 流程图符号

符 号	名 称	说 明
	起止框	表示算法的开始与结束
	判断框	用来根据给定的条件是否满足决定执行两条路径中的某一条路径
	处理框	用来表示赋值等一般操作
	输入/输出框	表示输入或输出操作
	流程线	表示流程的方向
	连接点	用于将没有画在一起的同一流程的各部分连接起来

用流程线将各种操作图符号连接在一起就构成了一个完整的算法流程图，例 1-1 中算法的流程图如图 1-1 所示。

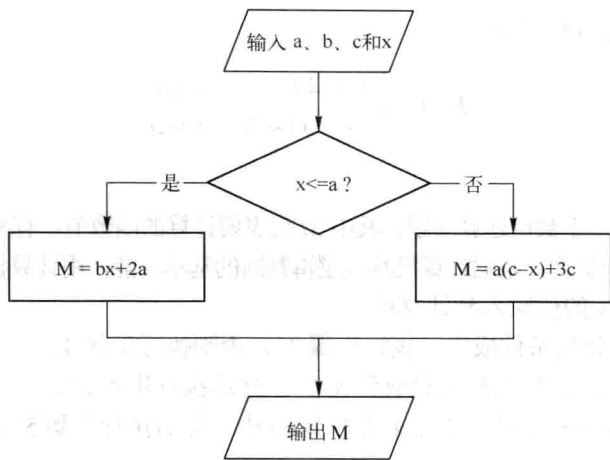


图 1-1 传统流程图

1.1.3 程序设计方法概述

用计算机解决工作中遇到的各种问题，常常需要设计和编写程序。程序的正确性、可靠性、可读性、可理解性、可修改性和可维护性如何，直接影响计算机的执行结果和使用效率，所以设计和编写程序并不是简单写一个程序，而是保证程序有很高的正确性、可靠性、可读性、可理解性、可修改性和可维护性。要达到这一目的，必须采用科学的程序设计方法。

程序设计方法种类很多，主要有模块化程序设计方法、结构化程序设计方法和面向对象程序设计方法等。

1. 模块化程序设计方法

模块化程序设计方法是一个常用且有效的方法。在设计和编写大型程序时，可以对其进行模块化分解，以降低程序的复杂性，提高程序的正确性、可靠性、可读性和可维护性。

模块是数据说明、接口声明和执行语句等程序对象的集合，可独立命名，并通过模块名来调用、访问和执行，如 VB 语言的子过程、函数、模块等程序对象可看成是模块。模块化就是把大程序划分成若干模块，每个模块完成一个子功能，模块间相互协调，共同完成特定功能，其实质是把复杂问题分解成许多容易解决的小问题，如图 1-2 所示。

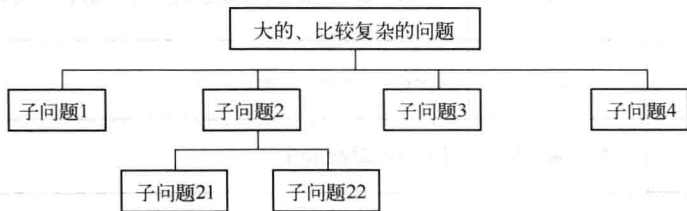


图 1-2 问题分解

值得注意的是，并不是模块分解得越细越好、模块数越多越好。实际上，当模块细化到一定程度时，因为模块数增加，其模块间接口复杂度和代价将增大。所以模块数不宜太多，应凭经验选择一个合适的模块数。

2. 结构化程序设计方法

结构化程序设计方法产生于 20 世纪 60 年代末，它对后来的程序设计方法的研究和发展产生了重大影响，直到今天它仍然是程序设计中采用的主要方法。

结构化程序设计的概念最早由著名计算机科学家 E. W. Dijkstra 提出。1965 年他在一次会议上指出：“可以从高级语言中取消 GOTO 语句”。1966 年，Bohm 和 Jacopini 证明了“只用三种基本的控制结构就能实现任意单入口和单出口的程序”。这三种基本控制结构是“顺序结构”“判断结构”和“循环结构”，如图 1-3 所示。1972 年，IBM 公司的 Mills 进一步提出，程序应该只有一个入口和一个出口。1971 年，IBM 公司在纽约时报信息库管理系统的设计中首次成功地使用了结构化程序设计技术。

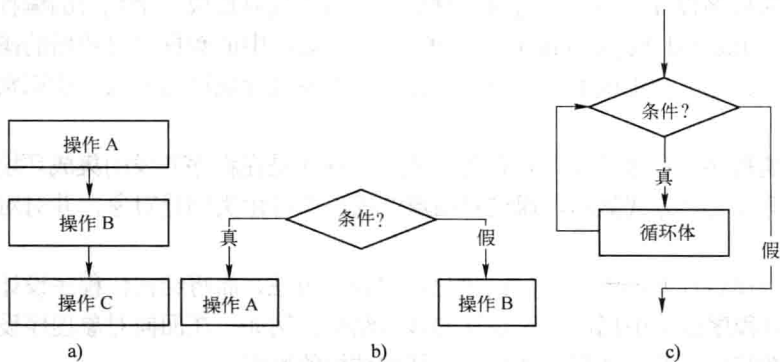


图 1-3 三种基本控制结构的流程图

a) 顺序结构 b) 判断结构 c) 循环结构

结构化程序设计主要包括：一是使用三种基本控制结构，二是采用自顶向下和逐步求精方法。结构化程序设计强调程序设计风格和程序结构的规范化，自顶向下和逐步求精方法是求解复杂问题的有效方法。自顶向下和逐步求精方法是由抽象到具体、由粗到细的方法。第一次细化称为“顶层设计”，然后通过一步一步细化，它们依次称为第二层、第三层设计，直到不需细化为止。细化结果可得到一个树型层次结构图，如图 1-4 所示。

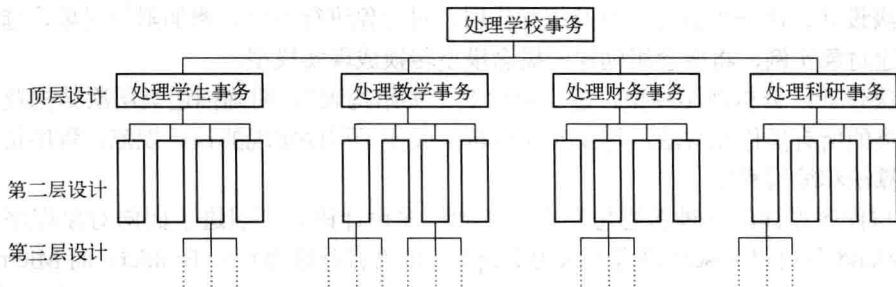


图 1-4 层次结构图

3. 面向对象程序设计方法

模块化和结构化程序设计方法属于传统的程序设计方法。进入 20 世纪 80 年代，随着计算

机科学技术的飞速发展，传统程序设计方法的不足和缺陷已经开始显现。传统的程序设计方法采用以功能和操作为驱动的思维模式，程序中数据从属于操作，该方法不利于人们有效地分析问题，算法和程序结构与求解的实际问题不完全一致，这种不一致性，大大降低了程序的可读性、可维护性和可修改性。用传统程序设计方法开发的程序不易维护和修改，一旦需要修改，将牵一发而动全身，修改工作量将会大幅度增加。针对传统程序设计方法的不足和缺陷，在 20 世纪 80 年代，开始研究面向对象程序设计方法。进入 20 世纪 90 年代，面向对象程序设计方法趋于完善和成熟，同时基于面向对象的程序设计语言不断出现，高级语言开始引入面向对象概念和理念，如 C++ 语言。

面向对象的程序设计方法是当前程序设计的大势所趋。面向对象的程序设计方法是对结构化程序设计方法的重新认识。在程序的设计中，算法总是与特定的数据结构密切相关，算法含有对数据结构的访问，特定的算法只适用于特定的数据结构，因此算法与数据结构在编程中应该是一个密不可分的整体，这个整体叫对象。

面向对象的程序设计通过类、对象、继承和多态等机制形成一个完善的编程体系。面向对象编程（Object-Oriented Programming, OOP）将程序设计中的数据与对数据的操作作为一个不可分割的整体，通过由类生成的对象来组织程序。对象包含属性与方法，能识别和响应一定的事件。

面向对象的程序设计多采用可视化的方式。可视化是在程序开发的集成环境中，将类与对象以可见的图形及文字方式显示，通过对图形的操作即可由类创建对象，并对对象的属性值进行操作。

面向对象程序设计并不绝对排斥结构化程序设计方法，而将结构化程序设计方法中的三种基本结构变为其程序设计中局部代码设计的基本结构。例如，在面向对象程序设计中，对象的事件方法、属性等代码的设计仍然遵循三种基本结构的原则。

用面向对象方法设计和编写程序，其结构与求解的实际问题完全一致，有很高的可读性、可维护性和可修改性。

面向对象程序设计方法一般分三级设计：概念级、系统级和规范级。

概念级设计：从实际问题出发，分析用户需求和功能需求，识别问题中所涉及的所有对象（实体）及相互关系，如学生管理中的学生名册、成绩单、学籍档案等，根据分析结果建立求解问题的概念模型（用图形表示）。详细描述每一对象的属性（一组变量、数据结构、状态）和操作（置初值、查询、修改、运算）。

系统级设计：进一步分析对象及相互作用，对对象进行取舍，增加附加对象，选择控制流方法，创建对象实例，将概念级创建的概念模型转换成现实模型。

规范级设计：分析现实模型，建立和绘制“类结构表”，明确描述类层次结构及其继承关系，描述类的所有操作和方法，将现实模型转换成便于程序实现的设计规范，程序员根据设计规范设计算法和编写程序。

为了与程序设计的这种新思想配合，各种程序设计语言都引进了面向对象程序设计的概念，如 Borland 公司将 Pascal 语言升级为 Delphi，C 语言升级为 C++ Builder；而 Microsoft 公司则将 Basic 升级为 Visual Basic。实际上，Visual Basic 语言因为其中导入事件驱动、面向对象、可视化等程序设计概念，功能已非常强大，用户可用它方便地完成从界面设计、数据库处理到多媒体控制等大部分任务。

1.2 Visual Basic 简介

Visual Basic (简称 VB) 是美国微软 (Microsoft) 公司推出的 Windows 环境下的软件开发工具, 使用 VB 可以既快又简单地开发 Windows 应用软件。

1.2.1 Visual Basic 概述

Visual 是指开发图形用户界面 (GUI) 的方法。Visual 的意思是“视觉的”或“可视的”, 也就是直观的编程方法。在 VB 中引入了控件的概念, 如各种各样的按钮、文本框、复选框等。VB 把这些控件模式化, 并且每个控件都由若干属性来控制其外观及工作方法。这样, 采用 Visual 方法无须编写大量代码去描述界面元素的外观和位置, 而只要把预先建立的控件加到窗体上, 就像使用“画图”之类的绘图程序, 通过选择画图工具来画图一样。

Basic 是指 BASIC (Beginner's All-purpose Symbolic Instruction Code) 语言, 之所以叫做“Visual Basic”, 就是因为它使用了 BASIC 语言作为代码。VB 在原有 BASIC 语言的基础上进一步发展, 至今包含了数百条语句、函数及关键词, 其中很多与 Windows GUI 有直接关系。VB 与 BASIC 有着千丝万缕的联系, 如果学过 BASIC 语言的话, 看到 VB 的程序结构会感到很亲切。专业人员可以用 VB 实现其他任何 Windows 编程语言的功能, 而初学者只要掌握几个关键词就可以建立实用的应用程序。

随着微型计算机技术的飞速发展, 美国微软公司的 Microsoft Windows 以其具有多任务性、图形用户界面、动态数据交换、对象链接与嵌入等强大功能, 而成为当今微型计算机操作系统的主流产品。众多的软件开发已从原来的 DOS 软件开发转向 Windows。许多商用软件公司为适应这一趋势推出了不少 Windows 环境下的软件开发工具, 如 Visual C++、Visual Basic、C++、Builder、Delphi、C# Builder PowerBuilder 等。但对于希望在 Windows 环境中开发一般的应用程序的初学者, VB 无疑是较理想的。使用 VB 不仅可以感受到 Windows 带来的新技术、新观念和新的开发方法, 而且 VB 是目前众多 Windows 软件开发工具中效率很高的一个。另外 VB 系列产品得到了计算机工业界的承认, 得到了许多软件开发商的大力支持。

1.2.2 Visual Basic 的发展过程

早在 1991 年, 为了简化 Windows 应用程序的开发, 微软公司推出了 Visual Basic 1.0。比尔·盖茨说, Visual Basic 1.0 的推出是“惊世骇俗的”“令人震惊的新奇迹”。它极大地改变了人们对 Windows 的看法以及使用 Windows 的方式。

1992 年, 经过对 Visual Basic 1.0 的修改后, 微软推出了 Visual Basic 2.0。

1993 年经再次修改完善后, Visual Basic 3.0 上市了。从这一版开始, Visual Basic 在 Windows 中几乎是无所不能的了!

1995 年, 随着 Windows 95 轰轰烈烈的发布, Visual Basic 4.0 也随之问世。

1997 年, 微软公司开始推出 Windows 开发工具套件 Microsoft Visual Studio 1.0, 其中包括了 Visual Basic 5.0, 1998 年发布的 Microsoft Visual Studio 98 则包含了 Visual Basic 6.0。

Visual Basic 5.0 以前的版本主要应用于 DOS 和 Windows 3.x 环境中 16 位程序的开发, 从 Visual Basic 5.0 以后的版本则只能运行在 Windows 95 或 Windows NT 操作系统下, 是一个 32 位应用程序的开发工具。

Visual Basic 6.0 共有 3 个版本：标准版、专业版、企业版。标准版主要是为初学者了解基于 Windows 的应用程序开发而设计的；专业版主要是为专业人员创建客户/服务器应用程序而设计的；企业版则是为创建更高级的分布式、高性能的客户/服务器或 Internet/Intranet 上的应用程序而设计的。

2002 年，微软公司发布 Visual Basic.NET，2003 年发布 Visual Basic 2003，2005 年发布 Visual Basic 2005，2008 年发布 Visual Basic 2008，2010 年发布 Visual Basic 2010，2012 年发布 Visual Basic 2012，2013 年发布最新版本 Visual Basic 2013。

1.2.3 Visual Basic 的特点

VB 是从 BASIC 发展而来的，对于开发 Windows 应用程序而言，VB 是目前所有开发语言中最简单、最容易使用的语言。作为程序设计语言，VB 程序设计具有以下特点。

1. 可视化的设计平台

用传统程序设计语言编程时，需要通过编程计算来设计程序的界面，在设计过程中看不到程序的实际显示效果，必须在运行程序的时候才能观察。如果对程序的界面不满意，还要回到程序中去修改，这一过程常常需要反复多次，大大影响了编程的效率。VB 提供的可视化设计平台，把 Windows 界面设计的复杂性“封装”起来。程序员不必再为界面的设计而编写大量程序代码，只需按设计的要求，用系统提供的工具在屏幕上“画出”各种对象，VB 自动产生界面设计代码，程序员所需要编写的只是实现程序功能的那部分代码，从而大大提高了编程的效率。

2. 面向对象的设计方法

VB 采用面向对象的编程方法（OOP），把程序和数据封装起来作为一个对象，并为每个对象赋予相应的属性。在设计对象时，不必编写建立和描述每个对象的程序代码，而是用工具“画”在界面上，由 VB 自动生成对象的程序代码并封装起来。

3. 事件驱动的编程机制

VB 通过事件执行对象的操作。在设计应用程序时，不必建立具有明显开始和结束的程序，而是编写若干个微小的子程序，即过程。这些过程分别面向不同的对象，由用户操作引发某个事件来驱动完成某种特定功能，或由事件驱动程序调用通用过程执行指定的操作。

4. 结构化的设计语言

VB 是在结构化的 BASIC 语言基础上发展起来的，加上了面向对象的设计方法，因此是更具有结构化的程序设计语言。

5. 充分利用 Windows 资源

VB 提供的动态数据交换（DDE）编程技术，可以在应用程序中实现与其他 Windows 应用程序建立动态数据交换、在不同的应用程序之间进行通信的功能。

VB 提供的对象链接与嵌入（OLE）技术则是将每个应用程序都看做一个对象，将不同的对象链接起来，嵌入到某个应用程序中，从而可以得到具有声音、影像、图像、动画及文字等各种信息的集合式文件。

VB 还可以通过动态链接库（DLL）技术将 C/C++ 或汇编语言编写的程序加入到 VB 的应用程序中，或是调用 Windows 应用程序接口（API）函数，实现软件开发包（Software Development Kit, SDK）所具有的功能。

6. 开放的数据库功能与网络支持

VB 具有很强的数据库管理功能。不仅可以管理 MS Access 格式的数据库，还能访问其他